Design and Analysis of Algorithms Assignment 6

Harrison Lee, Alex Zhao

March 29, 2019

Problem 7.28. *Office Hours Scheduling*

Given: I_1, I_2, ..., I_k intervals for office hours with some combination of available TAs for them, a to b office hours per TA, and c total office hours, find:

a) A polynomial-time algorithm that takes the input and constructs a valid schedule specifying which TAs cover which time slots, or reports there is no valid schedule.

b) The same algorithm, but with d_i office hours on given days i.

Algorithm 1. *a) Determine in polynomial time a valid schedule.*

First, set each interval I_1 to I_k as a node. Connect the interval nodes to a sink_1 node with edges of capacity 1 going from the interval node to the sink node, so that only 1 TA is assigned to each time slot. Connect this sink_1 node to a sink_2 node with an edge going from sink_1 to sink_2 with a minimum flow of c and capacity of c, to ensure exactly c office hours are scheduled.

For each TA, create a TA_source node and connect it to each time interval they can fill with an edge going from the TA to the time interval, setting the maximum capacity of these edges to 1. Create an overall source node and connect it to each TA node with edges going from the source to TA node, with minimum flows of a and capacities of b to ensure each TA has between a and b hours scheduled.

Run the Ford-Fulkerson network flow algorithm over this graph, as done in lecture, using the final flows as a schedule. To convert the flow to a schedule, each TA is scheduled for each time interval they flow through. Flow represents the number of hours scheduled to the time interval it passes through, and is assigned to the TA is flows from when it passes through the TA nodes.

Proof: We know it is a valid schedule if max flow is equal to c. The capacities and min-flows on the edges between source and each TA node will ensure that each TA has between a and b hours scheduled. Only 1 TA can be scheduled for each time interval due to the edges exiting each interval node having only a capacity of 1. The edge between sink_1 and sink_2 will ensure exactly c hours are scheduled each week. If no flow is returned or the max-flow is less than c, no schedule is possible.

b) Adjust the graph in part a) so that each interval connects to a node representing the day it is on, day_i, instead of the sink_1 node. These edges from the interval to day will still have capacities of 1. Connect each day_i node to the sink_1 node with edges going from day to sink node with infinite capacity and a minimum flow of d_i or 0, if no d is defined for that day.

Ford-Fulkerson can now be run like before on this graph, providing the final schedule in the same way. We know it is a valid schedule if max-flow is equal to c.

Proof: We know this schedule accommodates the required d_i office hour densities because the min-flows exiting each day node ensures at least d_i hours are scheduled and will flow through the intervals on day i.

Run Time: Run time is polynomial because setting up this graph has TA number of nodes, k interval nodes, and at most 7 day nodes, in addition to 3 source and sink nodes. There are k edges attached to each interval node connecting them to sink$_1$, up to $k * numberTAs$ edges connecting the TAs to interval nodes, 1 additional edge for each TA and day$_i$ node attaching them to source and sink$_1$ nodes respectively, with 1 additional edge connecting sink$_1$ and sink$_2$. This gives a total of Additionally Ford-Fulkerson runs in and adds polynomial time.

Problem 7.45. *Free-Standing Countries*

Given: A set of n countries with each country i having a budget surplus s_i. For each pair of countries i, j, there is the total exports from country i to j, e_{ij}

Find: If there exists a non-empty free-standing subset of the countries that is not the set of all countries.

We will reduce this problem to the min-cut problem in a fashion similar to the way we reduced project selection. Create a graph G with a node for each country and edges $e = (u, v)$ where $c(e) = e_{ij}$. Create a source node s with edges to all nodes i where $s_i > 0$ with capacity s_i and a sink node t with edges from all nodes i where $s_i < 0$, again with capacity s_i.

Then for an s-t cut, let S be all the nodes in A (except the source). Show that the capacity of the cut $c(A, B) = C - N$ where C is some constant and $N = \sum_{i \in A} s_i - \sum_{i \in A, j \in B} e_{ij}$ (which must be non-negative for S to be freestanding. By showing this relation, we know that if the capacity of the min cut $c(A, B) \leq C$, then the subset S is a free-standing subset (if the capacity of any valid cut is less than C, then the capacity of the min cut is also $\leq$ C, if the capacity of the min cut is $>$ C, then there is no other cut that could have capacity $\leq$ C. (A and B also need to each contain one node that isn't s or t so that S is neither empty nor the set of all countries)

Proof that $c(A, B) \leq C \rightarrow$free-standing subset:

$$c(A, B) = \sum_{i \in B, s_i > 0} c(s, i) + \sum_{i \in A, s_i < 0} c(i, t) + \sum_{i \in A, j \in B} e_{ij}$$

$$c(A, B) = \sum_{i \in B, s_i > 0} s_i + \sum_{i \in A, s_i < 0} -s_i + \sum_{i \in A, j \in B} e_{ij}$$

$$c(A, B) = \sum_{i \in B, s_i > 0} s_i - \sum_{i \in A, s_i < 0} s_i + \sum_{i \in A, j \in B} e_{ij} + \Big[\sum_{i \in A, s_i > 0} s_i - \sum_{i \in A, s_i > 0} s_i \Big]$$

$$c(A, B) = \sum_{i, s_i > 0} s_i - \sum_{i \in A} s_i + \sum_{i \in A, j \in B} e_{ij}$$

$$c(A, B) = C - \Big[\sum_{i \in A} s_i - \sum_{i \in A, j \in B} e_{ij} \Big] \textit{ where } C = \sum_{i, s_i > 0} s_i \; (\textit{a constant})$$

Proof that free-standing subset $\rightarrow c(A, B) \leq C$:

Same as above, by our construction of G.